

REMARKS

Applicant respectfully requests reconsideration in light of the arguments provided in this instant response. Claims 1 and 3-25 are pending. Claims 1 and 3-25 stand rejected.

CLAIM 26

Applicant respectfully notes that the Office Action mailed March 2, 2011 does not address Claim 26. Accordingly, Applicant understands Claim 26 to be allowable subject matter.

IMPROPER TO MAKE NEXT OFFICE ACTION FINAL

If the Office Action did intend to reject Claim 26, then Applicant respectfully submits that the Office Action mailed March 2, 2011 is not complete as to all matters and it would be improper to make the next Office Action final. For example, "The examiner's action will be complete as to all matters, except that in appropriate circumstances, such as mis-joinder of invention, fundamental defects in the application, and the like, the action of the examiner may be limited to such matters before further action is made" (emphasis added; 37 CFR §1.104(b)). "In rejecting claims for want of novelty or for obviousness, the examiner must cite the best references at his or her command" (37 CFR §1.104(c)(2)). In particular, "[w]here a claim is refused for any reason relating to the merits thereof it should be "rejected" and the ground of rejection fully and clearly stated" (emphasis added; MPEP §707.07(d)). Applicants respectfully submit that the Office Action mailed October 5th, 2009 did not respond to Applicants' remarks. For example, the Office Action mailed October 5th, 2009 did not respond to Applicants' remarks concerning the lack of motivation to combine, which starts on page 16 and proceeds through the second paragraph on page 17. Therefore, it would be improper for the next Office Action to be made final.

35 U.S.C. 101

Claims 17-25 stand rejected under 35 U.S.C. 101. On January 26, 2010, the USPTO published guidance on the Subject Matter Eligibility of Computer Readable Media, which recites in part (emphasis added):

The United States Patent and Trademark Office (USPTO) is obliged to give claims their broadest reasonable interpretation consistent with the specification during proceedings before the USPTO. See *In re Zletz*, 893 F.2d 319 (Fed. Cir. 1989) (during patent examination the pending claims must be interpreted as broadly as their terms reasonably allow). The broadest reasonable interpretation of a claim drawn to a computer readable medium (also called machine readable medium and other such variations) typically covers forms of non-transitory tangible media and transitory propagating signals *per se* in view of the ordinary and customary meaning of computer readable media, particularly when the specification is silent. See MPEP 2111.01. When the broadest reasonable interpretation of a claim covers a signal *per se*, the claim must be rejected under 35 U.S.C. § 101 as covering non-statutory subject matter. See *In re Nuijten*, 500 F.3d 1346, 1356-57 (Fed. Cir. 2007) (transitory embodiments are not directed to statutory subject matter) and *Interim Examination Instructions for Evaluating Subject Matter Eligibility Under 35 U.S.C. § 101*, Aug. 24, 2009; p. 2.

In particular, Applicant respectfully notes that the guidance specifically requires that claims be given “their broadest reasonable interpretation consistent with the specification” (emphasis added).

Applicant notes that the instant Office Action rejects Claims 17-25 under the assertion that it is within the scope of the disclosure that Claims 17-25 are directed toward a transitory propagating signal *per se*, and are thus non-statutory. However, Applicant respectfully submits that the specification describes only statutory embodiments of a computer readable medium and is silent with respect to any non-statutory embodiments of a computer readable medium. Therefore, Applicant respectfully submits that Claims 17-25, when reasonably interpreted consistent with the specification, are directed toward statutory subject matter, and thus overcome the instant rejection under 35 U.S.C. § 101.

35 U.S.C. 103

Claims 1 and 3-26 stand rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,125,447 by Gong (referred to herein as “Gong”) in view of “The C Book—Structures” (referred to herein as “C Book—Structures”). Applicant has reviewed the asserted art and respectfully submits that the asserted art does not teach or suggest Claims 1 and 3-26 for at least the following reasons.

“As reiterated by the Supreme Court in KSR, the framework for the objective analysis for determining obviousness under 35 U.S.C. 103 is stated in *Graham v. John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966). Obviousness is a question of law based on underlying factual inquiries” including “[a]scertaining the differences between the claimed invention and the prior art” (MPEP 2141(II)). “In determining the differences between the prior art and the claims, the question under 35 U.S.C. 103 is not whether the differences themselves would have been obvious, but whether the claimed invention as a whole would have been obvious” (emphasis in original; MPEP 2141.02(I)).

Applicant notes that “[t]he prior art reference (or references when combined) need not teach or suggest all the claim limitations, however, Office personnel must explain why the difference(s) between the prior art and the claimed invention would have been obvious to one of ordinary skill in the art” (emphasis added; MPEP 2141(III)).

Applicant respectfully submits that “[i]t is improper to combine references where the references teach away from their combination” (emphasis added; MPEP 2145(X)(D)(2); *In re Grasselli*, 713 F.2d 731, 743, 218 USPQ 769, 779 (Fed. Cir. 1983)). Applicant respectfully notes that “[a] prior art reference must be considered in its entirety, i.e., as a whole, including portions that would lead away from the claimed invention” (emphasis in original; MPEP 2141.02(VI); *W.L. Gore & Associates, Inc. v. Garlock, Inc.*, 721 F.2d 1540, 220 USPQ 303 (Fed. Cir. 1983), *cert. denied*, 469 U.S. 851 (1984)). Further, “[a] reference will teach

away if it suggests that the line of development flowing from the reference's disclosures is unlikely to be productive of the result sought by the applicant. *In re Gurley*, 31 USPQ2d 1130 (Fed. Cir. 1994)."

Applicant respectfully submits that Gong does not teach or suggest "wherein said portions of code are not required to be associated with one or more object oriented classes," as recited by independent Claim 1, as will become more evident. Further, Applicant respectfully submits that Gong teaches away from "wherein said portions of code are not required to be associated with one or more object oriented classes," as recited by independent Claim 1, as will become more evident.

For example, Gong states at Col. 1 lines 60-62, "Recently developed methods of running applications involve the automatic and immediate execution of software code loaded from remote sources over the network." Gong further states at Col. 2 lines 44-50,

Based on the foregoing, it is clearly desirable to develop a method which reduces the effort and in-depth knowledge required to modify permissions established for the sources of code being executed by a computer system. It is further desirable to develop a method which reduces the effort and in-depth knowledge required to create new permissions.

Accordingly, Applicant understands Gong's intended purpose is to reduce the effort and in-depth knowledge required to modify permissions or to create new permissions to provide security for the code that, for example, may be loaded from remote resources over a network (see Gong Col. 2 lines 44-50; Col. 1 lines 60-62 quoted herein). As will become more evident, Applicant understands Gong's code to be instances of OO classes.

Applicant respectfully submits that Gong achieves Gong's intended purpose by mapping OO classes to protection domains (see Gong abstract; Col. 2 line 66- Col. 3 line 1; Col. 6 lines 44-45; Col. 6 lines 55-62; Col. 8 lines 39-43; Col. 10 lines 49-51; Col. 10 lines 56-57, Col. 12 lines 39-47 quoted herein). For

example, Gong states in the last sentence of the abstract, "When an object requests an action, a determination is made as to whether the action is permitted based on the class to which the object belongs and the association between classes and protection domains." Gong states at Col. 2 line 66 to Col. 3 line 1, "For example, based on policy data, an association between Class CA and protection domain PA, and class CB and protection domain PB is established." Gong states at Col. 6 lines 44-45, "...without requiring specialized knowledge of complex security-management techniques." Gong states at Col. 6 lines 55-62,

Code executor 210 executes code which code executor 210 receives from code stream 220. One example of a code executor is a Java virtual machine. A Java virtual machine interprets code called byte code. Byte code is code generated by a Java compiler from source files containing text. The Java virtual machine is described in detail in Tim Lindholm & Frank Yellin, *The Java Virtual Machine Specification* (1996).

Further, Gong states at Col. 8 lines 39-43, "According to an embodiment of the present invention, protection domains are used to enforce security within a computer system. A protection domain can be viewed as a set of permissions granted to one or more principles." Gong states at Col. 10 lines 49-51, "Then protection domain object 282 is created and populated with the permission just mentioned." Gong states at Col. 10 lines 56-57, "Next, in step 428, the mapping of the class to the protection domain is established." Gong states at Col. 10 lines 63-64, "Creating an association of classes to protection domains in the manner just described offers several advantages." Gong states at Col. 12 lines 39-47,

Examining the permissions of a particular protection domain associated with an object begins by determining an object's class. A code executor, such as a Java Virtual machine, provides that each object incorporates a method which returns the class of an object. In this example, the first object with a method on the call stack is object a. Access controller 280 invokes the method that returns the class of object a.

Accordingly, Applicant understands Gong to require his code to be instances of object oriented (OO) classes so that Gong can easily map Gong's protection domain, which represent Gong's protection attributes, to the OO classes (see Gong abstract; Col. 2 line 66- Col. 3 line 1; Col. 6 lines 44-45; Col. 6

lines 55-62; Col. 8 lines 39-43; Col. 10 lines 49-51; Col. 10 lines 56-57, Col. 12 lines 39-47 quoted herein).

By mapping OO classes to protection domains (see Gong abstract; Col. 2 line 66- Col. 3 line 1; Col. 6 lines 44-45; Col. 6 lines 55-62; Col. 8 lines 39-43; Col. 10 lines 49-51; Col. 10 lines 56-57, Col. 12 lines 39-47 quoted herein), Applicant understands Gong to achieve Gong's intended purpose of reducing the effort and in-depth knowledge required to modify permissions or to create new permissions to provide security for the code (e.g., the instances of the OO classes) that, for example, may be loaded from remote resources over a network (see Gong Col. 2 lines 44-50; Col. 1 lines 60-62 quoted herein). Therefore, Applicant understands Gong to require his code to be instances of OO classes.

In contrast, Claim 1 recites "wherein said portions of code are not required to be associated with one or more object oriented classes." Applicant respectfully submits that requiring code to be instances of OO classes teaches away from "wherein said portions of code are not required to be associated with one or more object oriented classes," as recited by Claim 1. Therefore, Applicant respectfully submits that Claim 1 is patentable over Gong.

Further, since Applicant understands Gong to teach away from "wherein said portions of code are not required to be associated with one or more object oriented classes," as recited by Claim 1, Applicant respectfully submits that there is no motivation to combine Gong with any other asserted art, such as C Book—Structures. Therefore, Applicant respectfully submits that Gong does not teach or suggest Claim 1 and even teaches away from Claim 1.

Applicant respectfully submits that C-Book Structures does not remedy the deficiencies in Gong. For example, the title of C-Book—Structures is "The C Book—Structures." As any one of ordinary skill in the art understands, C is a purely non-object oriented language. However, as discussed herein, Applicant understands Gong to require his code to be instances of OO classes. Since

Gong requires his code to be instances of OO classes, Applicant respectfully submit that modifying Gong to use a purely non-object oriented language would render Gong in operable for Gong's intended purpose. Therefore, Applicant respectfully submits that there is no motivation to combine Gong and C Book—Structures because Applicant understands Gong and C Book—Structures to teach away from each other. Further, Applicant respectfully submits that there is no motivation to combine Gong with any other asserted art, such as C Book—Structures, because as discussed herein, Applicant understands Gong to teach away from "wherein said portions of code are not required to be associated with one or more object oriented classes," as recited by Claim 1.

The Office Action states on page 4 lines 3-9,

Gong's disclosure is limited to explicitly discussing the preferred embodiment wherein all the pertinent software is implemented as Java objects, Java being a well-known object-oriented programming language with classes (col. 6, line 45 – col. 7, line 60). However, Gong merely assumes that the object oriented requirement is true (Ibid, particular col. 6, lines 65-66); yet his preferred embodiment is illustrative but not restrictive, and variations as to the specifics of how his invention is implemented are permitted (col. 13, lines 23-30).

Applicant respectfully disagrees that Gong is merely discussing a preferred embodiment.

First, Applicant shall discuss the portions of Gong (e.g., Col. 6 line 45 to Col. 7 line 60, Col. 6; lines 65-66, Col. 13 lines 23-30) that the Office Action referred to on page 4 lines 3-9 of the Office Action.

The Office Action appears to assert on page 4 lines 4-6 that Gong indicates that Java is a well known object-oriented programming language with classes at Col. 6 line 45 to Col. 7 line 60. Applicant agrees that Gong indicates that Java is a well known object-oriented programming language with classes.

At Col. 6 lines 65-66, Gong states, "Consequently, the code is in the form of methods associated with objects that belong to classes." Accordingly, Applicants understand Gong to use object oriented classes to divide his code into portions that security can be provided to, as discussed herein.

Gong states at Col. 13 lines 23-30,

In the foregoing specification, the invention has been described with reference to specific embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

Applicant respectfully notes that Gong does not describe any "variations as to the specifics" at Col. 13 lines 23-30. Instead, Applicant understands Gong to provide boiler plate language indicating that modifications and changes may be made without departing from the broader spirit and scope of the invention at Col. 13 lines 23-30. Applicant understands Gong to describe all of Gong's embodiments in the context of object oriented classes and understands Gong to be silent with respect using non-object oriented languages for any of Gong's embodiments.

Further, as discussed herein, by mapping OO classes to protection domains (see Gong abstract; Col. 2 line 66- Col. 3 line 1; Col. 6 lines 44-45; Col. 6 lines 55-62; Col. 8 lines 39-43; Col. 10 lines 49-51; Col. 10 lines 56-57, Col. 12 lines 39-47 quoted herein), Applicant understands Gong to achieve Gong's intended purpose of reducing the effort and in-depth knowledge required to modify permissions or to create new permissions to provide security for the code (e.g., the instances of the OO classes) that, for example, may be loaded from remote resources over a network (see Gong Col. 2 lines 44-50; Col. 1 lines 60-62 quoted herein).

Therefore, Applicant respectfully disagrees that Gong is merely discussing a preferred embodiment. Instead, Applicant understands Gong to require his code that he provides security for to be instances of OO classes.

In yet another example, Gong states at Col. 2 lines 44-64 (emphasis added),

Based on the foregoing, it is clearly desirable to develop a method which reduces the effort and in-depth knowledge required to modify permissions established for the sources of code being executed by a computer system. It is further desirable to develop a method which reduces the effort and in-depth knowledge required to create new permissions.

SUMMARY OF THE INVENTION

A method and system are provided for implementing security policies within a computer system. The security mechanism makes use of structures referred to herein as “protection domains” to organize, represent and maintain the security policies that apply to the computer system.

According to one aspect of the invention, protection domains are established based on policy data, where each protection domain is associated with zero or more permissions. An association is established between the protection domains and classes of objects (i.e., instantiations of the classes) that may be invoked by the computer system. When an object requests an action, a determination is made as to whether the action is permitted for that object...

Accordingly, Applicant understands Gong to state at Col. 2 lines 44-50 that his intended purpose is to reduce the effort and in-depth knowledge required to modify permissions or to create new permissions to provide security for the code (e.g., the instances of the OO classes) that, for example, may be loaded from remote resources over a network (see Gong Col. 2 lines 44-50; Col. 1 lines 60-62 quoted herein). Further, Applicant notes that immediately after Gong states his intended purpose at Col. 2 lines 44-50 quoted herein, Gong teaches establishing an association between protection domains and classes of objects (i.e. instantiations of the classes) that may be invoked by the computer system (see Gong Col. 2 lines 60-63 quoted herein). Accordingly, Applicant understands

Gong to require his code to be instances of OO classes in order to achieve his intended purpose. Therefore, Applicants understand Gong to teach away from “wherein said portions of code are not required to be associated with one or more object oriented classes,” as recited by independent Claim 1.

Accordingly, Applicant respectfully submits that Claim 1 is patentable over the Gong C Book—Structures combination.

For similar reasons, Applicant respectfully submits that independent Claims 10, 12, 15 and 17 are patentable over the Gong C Book—Structures combination in that independent Claims 10, 12, 15 and 17 also recite “wherein said portions of code are not required to be associated with one or more object oriented classes.”

CONCLUSION

In light of the above listed amendments and remarks, reconsideration of the rejected claims is requested. Based on the arguments and amendments presented above, it is respectfully submitted that Claims 1 and 3-25 overcome the rejections of record. For reasons discussed herein, Applicant respectfully requests that Claims 1 and 3-25 be considered by the Examiner. Therefore, allowance of Claims 1 and 3-25 is respectfully solicited.

Should the Examiner have a question regarding the instant amendment and response, the Applicant invites the Examiner to contact the Applicant's undersigned representative at the below listed telephone number.

Respectfully submitted,
WAGNER BLECHER LLP

Dated: 04/28/2011

/John P. Wagner, Jr./

John P. Wagner, Jr.
Registration No. 35,398

Address: Westridge Business Park
123 Westridge Drive
Watsonville, California 95076 USA

Telephone: (408) 377-0500 Voice
(408) 234-3649 Direct/Cell
(831) 722-2350 Facsimile